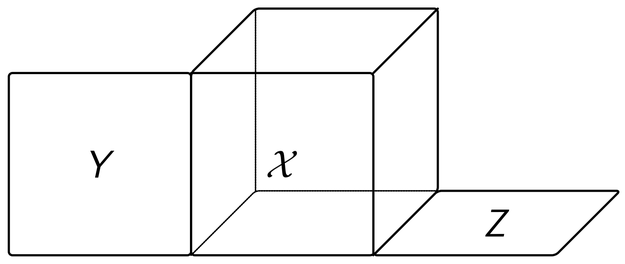
Structured data fusion

Laurent Sorber[1], Marc Van Barel[1] and Lieven De Lathauwer[2]

Structured data fusion



Structured data fusion (SDF) is a framework for the rapid prototyping of knowledge discovery in one or more (possibly incomplete) data sets:

- Each data set ($\mathcal{X}$, Y and Z) is represented as a tensor.
- Each tensor is factorized with a tensor decomposition.
- The factorizations are coupled with each other by indicating which factors should be shared.
- Factors may be imposed to have any type of structure which can be constructed as an explicit function of some underlying variables.

Example: Netflix $1,000,000 challenge

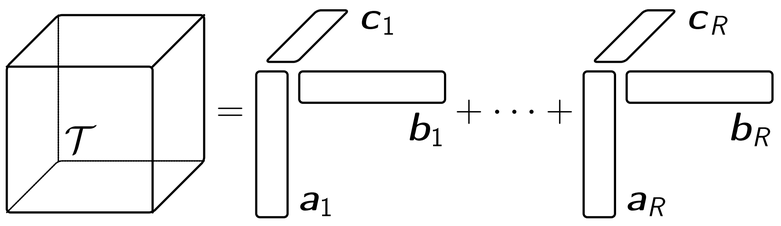
- $\mathcal{X}$ is a movie x date x user tensor of size 20,000 x 3,000 x 500,000 containing ratings between 1 and 5 stars.
- Y is a movie x director matrix with director information.
- Z is a user x user matrix with friendship links.
- *Objective:* fill in unknown ratings using these data sets.
- *Algorithm:* joint low rank approximation with SDF.

Coming soon: Tensorlab v2.0

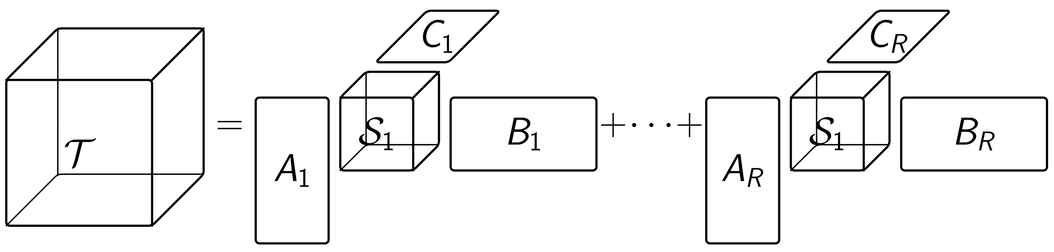
Tensorlab v2.0 offers a domain specific language for SDF.
See the examples on the right!

Tensorlab offers a library of tensor decompositions for SDF.
Adding your own tensor decompositions requires little effort!
The library currently includes:

- Canonical polyadic decomposition



- Block term decompositions



Tensorlab offers a library of factor structures for SDF.
Adding your own factor structures requires little effort!
The library currently includes:

- Nonnegativity
- Orthogonality
- Matrix inverse
- Toeplitz
- Hankel
- Vandermonde
- Band structured
- Lower/upper triangular
- Columns as continuous functions

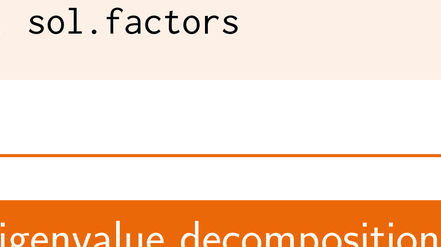
Example: QR factorization

In MATLAB:

$$A = Q \cdot R$$

```
[Q,R] = qr(A);
```

With Tensorlab's DSL for SDF:




```
model.variables.q = randn(N*(M-(N-1)/2),1);
model.variables.r = randn(N*(N+1)/2,1);

model.factors.Q = {'q', ...
@(x,y)struct_orth(x,y,[M N])};
model.factors.R = {'r',@struct_tril};

model.factorizations.qr.data = A;
model.factorizations.qr.cpd = {'Q','R'};

sol = sdf_nls(model);
sol.variables, sol.factors
```

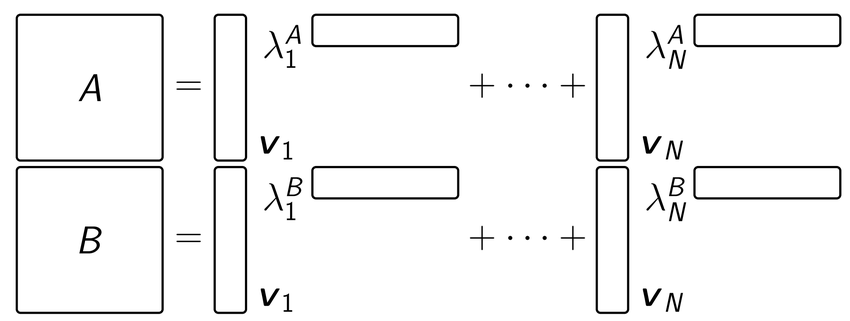
Example: joint eigenvalue decomposition

In MATLAB:

$$A = V \cdot \Lambda_A \cdot V^{-1}$$
$$B = V \cdot \Lambda_B \cdot V^{-1}$$

```
[V,DA] = eig(A);
DB = inv(V)*B*V;
```

With Tensorlab's DSL for SDF:




```
model.variables.v = randn(N,N);
model.variables.da = randn(1,N);
model.variables.db = randn(1,N);

model.factors.V = 'v';
model.factors.Vinv = {'v',@struct_invtransp};
model.factors.DA = 'da';
model.factors.DB = 'db';

model.factorizations.eigA.data = A;
model.factorizations.eigA.cpd = {'V','Vinv','DA'};
model.factorizations.eigB.data = B;
model.factorizations.eigB.cpd = {'V','Vinv','DB'};

sol = sdf_nls(model);
```

KU LEUVEN

[1]{Laurent.Sorber,Marc.VanBarel}@cs.kuleuven.be, [2]Lieven.DeLathauwer@kuleuven-kulak.be
Computer Science & Electrical Engineering Departments — Work Group 2: Data-driven modelling